

WestminsterResearch

<http://www.westminster.ac.uk/westminsterresearch>

Adaptive-threshold region merging via path scanning

G. Gupta and A. Psarrou

This is a copy of the author's accepted version of a paper subsequently published in the proceedings of the *22nd International Conference on Pattern Recognition* Stockholm, Sweden, Aug 2014, IEEE .

It is available online at:

<https://dx.doi.org/10.1109/ICPR.2014.173>

© 2014 IEEE . Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

The WestminsterResearch online digital archive at the University of Westminster aims to make the research output of the University available to a wider audience. Copyright and Moral Rights remain with the authors and/or copyright owners.

Whilst further distribution of specific materials from within this archive is forbidden, you may freely distribute the URL of WestminsterResearch: (<http://westminsterresearch.wmin.ac.uk/>).

In case of abuse or copyright appearing without permission e-mail repository@westminster.ac.uk

Adaptive-threshold region merging via path scanning

Gaurav Gupta

Faculty of Science and Technology
University of Westminster, London, UK
Email: guptaga@westminster.ac.uk

Alexandra Psarrou

Faculty of Science and Technology
University of Westminster, London, UK
Email: psarroa@westminster.ac.uk

Abstract—Region merging algorithms commonly produce results that are seen to be far below the current commonly accepted state-of-the-art image segmentation techniques. The main challenging problem is the selection of an appropriate and computationally efficient method to control resolution and region homogeneity. In this paper we present a region merging algorithm that includes a semi-greedy criterion and an adaptive threshold to control segmentation resolution. In addition we present a new relative performance indicator that compares algorithm performance across many metrics against the results from human segmentation. Qualitative (visual) comparison demonstrates that our method produces results that outperform existing leading techniques.

I. INTRODUCTION

Segmentation of an image into meaningful regions is an important task in computer vision. The state-of-the-art, with respect to both quality and speed of image segmentation, mainly revolves around the mean-shift [13] and graph based [16] algorithms. However, one of the most commonly applied family of techniques is region merging, where pixels are step-wise grouped into larger and larger segments.

Region merging, a form of agglomerative hierarchical clustering, lags behind the state-of-the-art because it most commonly applies a greedy merge mechanism which produces a lower quality segmentation caused by a tradeoff between resolution and accuracy [8]. Attempts to adaptively change the threshold or the merge criteria on the fly [10] have improved the results to an extent, but have not managed to match the best performing segmentation techniques. The problem may be attributed in part to the greedy nature of merging schemes. The best local merge is not guaranteed to be optimal in a global sense. However, greedy merging is what gives the technique its speed, and applying extended merging criteria [8] reduces this advantage.

In this paper we present a region merging segmentation method, employing both a semi-greedy merge criterion and an adaptive threshold. This segmentation method rivals the results of state-of-the-art algorithms both in quantitative and qualitative terms. The rest of the paper is organised as follows. Section II discusses the development of region merging as a segmentation method. Section III presents the region merging scheme along with post-processing to reduce the number of segments. Section IV shows quantitative and qualitative results. Finally, Section V concludes the paper.

II. BACKGROUND

Some of the leading approaches to image segmentation are graph or tree based ([16], [28]), probabilistic ([3], [11]) and

statistical ([13], [27], [12]). Another family of segmentation techniques, region merging, has had less success at matching the quality-speed tradeoff of the leading methods, however is still popularly used.

Region merging imposes direct topological constraints in which no two non-neighbouring regions may be merged. This ensures that topological proximity becomes a primary factor in the merge process. Some early work in region merging involved classical greedy merging using local information and the regions obtained were applied towards scene analysis [7]. The introduction of the technique of seeded region growing [2] performed image segmentation by establishing seed points in the image and then growing regions around these. The results were sensitive to the initial seed generation. While there has been effort on improving the seed generation process, the task of seed generation itself implies some knowledge of the image structure before segmentation is known.

Another important consideration found in region merging is the order in which pixels or regions are considered for merging. There has been work on the dependency between scan order and the segmentation results, with different paths for path-based labelling are discussed in [17], statistically-based reinterpretation in [26], and the entire region merging framework reanalysed in [8].

The region merging framework consists of three main components: the region model, the merging criterion, and the merging order. Some common merging criteria are given in [15] and [8] which also explores various options and combinations of these components. We proceed from this foundation by proposing a novel segmentation method, involving a fast but effective region merging scheme. In particular, our method favours a best merge over a fast merge. Our method additionally allows for multiple merges over each iteration as opposed to only a single merge such as in the hierarchical stepwise optimisation [6]. The maximum number of possible merges in a single pass is limited only by the image size and the homogeneity distribution.

III. SEMI-GREEDY ADAPTIVE-THRESHOLD REGION MERGING

The Beaulieu-Goldberg hierarchical stepwise optimisation (HSWO) algorithm [6] performs region merging on N clusters by calculating at each iteration the distance measures $C_{i,j} = d(S_i, S_j)$ for all cluster pairs (S_i, S_j) , with the most similar pair of clusters being merged. This merging procedure is repeated until some stopping criterion is satisfied.

A. Neighbourhoods and scan order

The method proposed in this paper may be thought of as a refinement of the basic HSWO algorithm, and thus similar performance considerations apply. As the authors note, while for N clusters there are $N \times (N - 1)$ cluster pairs, in region merging we consider only adjacent segments and the average number of neighbours per segment M is typically small ($4 \leq M \leq 8$), thus limiting the number of potential segment pairs per iteration to $N \times M$. They additionally note that a limited number of new segment pairs must be considered at each iteration since a segment merge only affects the adjacent segments.

The calculation of all relevant merge costs at each iteration is still an expensive operation, either in terms of computer memory if a list of merge costs are maintained or in terms of processing speed if some merge costs are recalculated at each step. With a suitable neighbourhood scanning procedure however, some further gains in resource utilisation may be made. It is common for example to use 4 or 8 connectivity when considering neighbouring pixels. For reasons of scan efficiency we use a rectangular L-shaped neighbourhood with a left to right and top to bottom scan path in our work.

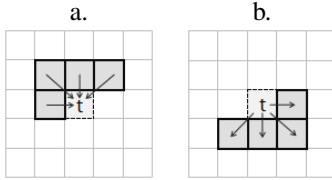


Fig. 1. Neighbourhoods at center pixel t for left to right, top to bottom scans. a) Seeking labels from $\{TL, T, TR, L\}$ neighbours, b) Pushing labels onto $\{R, BL, B, BR\}$ neighbours

We encode the eight possible neighbours for any pixel centered at pixel t as $\{TL, T, TR, L, R, BL, B, BR\}$, (top left, top, top right, left, right, bottom left, bottom and bottom right respectively). An L-shaped neighbourhood reduces scan redundancy while not missing any potential merge pathways. For computational efficiency and algorithmic correctness we wish to have the smallest neighbourhoods that exhibit two characteristics: a) the neighbourhood should contain either only already-labelled or only yet-unlabelled pixels, and b) the neighbourhood should not miss any possible merge pathways. The two possible *minimum* L-neighbourhoods for a row-wise scan are $\{TL, T, TR, L\}$ and $\{R, BL, B, BR\}$, see Figure 1. Given the usual row-wise nature of image scans, the former set is suited for *pull-labelling*, in which the pixel currently being considered seeks to adopt or pull a label from one of the L-neighbours, while the latter set is suited for *push-labelling*, in which the current pixel seeks to propagate or push its own label onto one or more of the L-neighbours. In our work we use the 4-neighbourhood $\{TL, T, TR, L\}$ in the row-wise scan to check for potential merges, and the 8-neighbourhood $\{TL, T, TR, L, R, BL, B, BR\}$ to check if the merge is the best merge for both involved segments.

The use of a single-pass left to right, top to bottom scan leads to some trivial oversegmentation. Looped region connectivity paths in rightward arcs at least partially unconnected after the first pass (segments 2 and 4 in Figure 2) and a second

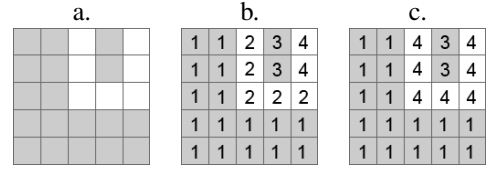


Fig. 2. Label assignment: a) Original image, b) After first pass, c) After second pass

pass is recommended to correct this. Using an iterative best-merge algorithm however circumvents this problem since any merges missed on one iteration will be automatically corrected on the next as long as the merge criterion stays the same or becomes more relaxed as the segmentation evolves.

B. Distance measure

One of the most important criterion for ensuring a good balance between accuracy and speed is the distance measure used to compute segment distances and thus their merge costs. The work by Mignotte [23] compares the performance of several common distance measures over segmentation results using their clustering method on the Berkeley Segmentation Dataset [21]. The measures compared are the Bhattacharya, Euclidean, Manhattan, Chord, Kolmogorov, Histogram intersect, Kullback, and Shannon-Jensen distances. Their results show that the Bhattacharya and Manhattan distances perform the best with respect to four important segmentation evaluation metrics, the PRI, VOI, GCE and BDE (see Section IV for our results over these metrics).

Given the much lower computational expense of the Manhattan distance relative to the Bhattacharya distance, in our work we exclusively use the Manhattan distance, $d(S_i, S_j) = \sum_{i=0}^n |U_i - V_i|$, where U_i and V_i are the feature vectors for segments S_i and S_j respectively, and n is the length the feature vector, which in our case is 3 from the $\{R, G, B\}$ colour channels.

C. Optimal local merges

While a per-iteration single merge is inefficient, using such an L-neighbourhood row-wise scan path leads to multiple local merges which are not guaranteed to be the best outside of the local region being considered. For a center pixel i , a potential best merge pair (S_i, S_j) may be suboptimal for pixel j , if there exists a neighbour k of j for which $C_{j,k} < C_{i,j}$ is true, indicating a better merge. A merge over any particular L-neighbourhood would therefore also involve the L-neighbourhood of the target of the merge. This semi-greedy behaviour ensures that a local best merge is desirable at a wider scale than the immediate L-neighbourhood, potentially even being a globally best merge and additionally still allowing multiple merges to take place at different image locations during every iteration through the image.

Additionally there is the matter of segmentation resolution, which is typically controlled in most region merging algorithms by some stopping criterion such as a distance threshold. This threshold determines the range of acceptable feature-space distance values for any two segments in order for a merge to take place between them. In almost all cases the feature distance $d(S_i, S_j)$ between two segments and the

cost $C_{i,j}$ of merging them can be defined equivalently, and in this paper we will refer interchangeably to them. Merges take place whenever the condition $d(S_i, S_j) < d_{max}$ holds true. Increasing the range of d_{max} lowers the segmentation resolution and leads to an undersegmentation while restricting its range has the opposite effect of increasing resolution and leading to an oversegmentation.

We use a very computationally simple scheme for adapting the distance threshold as the segmentation progresses, very similar to the *dynamic merge relaxation* in [31]. A large threshold allows incorrect merges to occur if two regions locally meet the merge criterion even if on a more global scale the merge in question would be suboptimal. This problem can be alleviated if the regions are given more time to move towards their respective cluster centers in feature space.

Based on this, we apply an inflation term to the merge threshold so that small regions of relatively low salience are merged as the threshold rises but highly salient small regions are not. We use a simple expanding threshold, which starts small and gradually grows towards the final threshold value. On the first few iterations the small threshold value ensures only the most locally correct merges take place, having the effect of shifting regions towards their feature-space cluster centers. As the threshold rises, these feature-shifted regions now merge only with their locally nearest-feature neighbours, thus mitigating the problem of incorrect merges had we started with a large distance threshold. The full segmentation is a two phase procedure consisting of the following:

- 1) Algorithmic region merging
- 2) Region reduction
 - a) Weakness heuristic region reduction
 - b) Small segment reduction
 - c) Enclosed region absorption

The region reduction phase is discussed in Section III-D.

Before we perform region merging, we convolve the image with a median filter to reduce any existing salt and pepper type noise. The median filter is represented as $y(m, n; W) = \text{med}\{x(m-k, n-l), (k, l) \in W\}$, where W in our case is a 3×3 window or filter mask.

We adapt and extend the HSWO algorithm from [6] by including a semi-greedy merge criterion and an adaptive inflating threshold. Some significant differences are the use of multiple merges over any iteration and the ability to change all labels for a given segment at any time and any number of times within an iteration (to correct possible labelling errors arising from local label-seeking that misses merge pathways in the blind spot of the L-neighbourhood). The key controller of the row-wise scan is the variable t which is the pixel index currently being considered. Incrementing t has the effect of moving through the image row-wise from left to right, the minimum L-shaped neighbourhood resulting directly from looking only at all the pixels already labeled in the past within the current iteration. The variables involved are as follows:

- 1) B_i , the set of the segments adjacent to S_i , called the neighborhood,
- 2) D_i , the parameters that describe the segment S_i , e.g. the segment R, G, B means,

3) $C_{i,j} = C(D_i, D_j)$, the cost of merging segment S_i with S_j , where S_j is contained in B_i , and

4) d_{curr} , the distance threshold that restricts merges if the cost of merging is greater than this value. The threshold grows after each iteration as $d_{curr} = d_{curr} + d_{step}$, where $d_{step} = 1$ for the slowest evolution of the segmentation map.

5) d_{max} , the maximum allowable distance threshold such that the following always holds: $d_{curr} \leq d_{max}$.

The region merging algorithm is then as follows:

- I. *Initialise:*
 - (i) $Ind = \{1, 2, \dots, n\}$ (image pixel indices).
 - (ii) $P^0 = \{S_1, S_2, \dots, S_n\}$ (initial partition).
 - (iii) $\text{Label}(t), t \in Ind$ (segment label for pixel t).
 - (iv) $k = 0, m = n$ and $d_{curr} = 0$.
 - (v) $\forall S_i \in P^0$, calculate D_i and B_i .
 - (vi) $hasMerged = false$
- II. *Merge, $\forall t \in Ind$, 4-neighbour scan $\{TL, T, TR, L\}$:*
 - (i) $i = \text{Label}(t)$.
 - (ii) calculate $CS_i = \{C_{i,j} | S_j \in B_i\}$.
 - (iii) find $C_{u,v} = \text{Minimum}(C_{i,j})$ where $C_{i,j} \in CS_i$, and $\text{Minimum}(C_{v,t}) \geq C_{u,v}$ where $C_{v,t} \in CS_v$ in a full 8-neighbourhood $\{TL, T, TR, L, R, BL, B, BR\}$.
 - (iv) if $C_{u,v} \leq d_{curr}$, do $\text{Merge}(S_u, S_v)$ as follows:
 - a) $k = k + 1$ and $m = m + 1$.
 - b) $P^k = (P^{k-1} \cup \{S_m\}) \cap \{S_u, S_v\}$.
 - c) calculate D_m from D_u and D_v .
 - d) $B_m = (B_u \cup B_v) \cap \{S_u, S_v\}$.
 - e) $\forall S_j \in B_m, B_j = (B_j \cup \{S_m\}) \cap \{S_u, S_v\}$
 - f) $hasMerged = true$
- III. *Stopping condition:*
 - (i) if $hasMerged == true$, do the following:
 - a) $hasMerged = false$.
 - b) $d_{curr} = \text{Min}(d_{curr} + d_{step}, d_{max})$.
 - c) go to step II.
 - (ii) stop.

D. Further region reduction

Two examples of heuristics applied in previous work in order to reduce the number of segments are the phagocyte heuristic and the weakness heuristic [7]. The phagocyte heuristic acts so as to smoothen or shorten region boundaries while the weakness heuristic joins regions based on the strength of the boundary that separates them. The phagocyte heuristic is less general since objects in real world images are not always guaranteed to have smooth boundaries. The weakness heuristic on the other hand is more general with respect to real world images in the sense that similar segments separated by a weak boundary are likely to belong together, and thus may be merged.

We employ the weakness heuristic in our algorithm to clean up the segmentation and reduce the number segments.

Although a larger distance threshold in the main algorithm would also result in fewer segments, noise in the data may cause two dissimilar segments to be joined since merging may occur over any two adjacent pixels. The weakness heuristic is more robust since similarity is gauged over the length of the boundary separating two regions, thus the decision to merge or not will be less affected by a little noise over the boundary pixels.

The variables involved in the region reduction post-processing are as follows:

1) for two regions S_i and S_j where $S_j \in B_i$, let $L_i = \{l_{i,1}, l_{i,2}, \dots, l_{i,q}\}$ and $L_j = \{l_{j,1}, l_{j,2}, \dots, l_{j,r}\}$ be the set of all boundary pixels in S_i and S_j respectively.

2) also let $\text{Adjacent}(l_{i,u}, l_{j,v})$ be a boolean function determining whether pixels $l_{i,u} \in L_i$ and $l_{j,v} \in L_j$ are immediately adjacent to each other. Note that this adjacency function finds three adjacent pixels in S_j for every pixel in S_i along smooth boundaries, which we will need to compensate for in the boundary strength computation.

3) then $L_i^j \subset L_i$, where $\exists u \in L_i, v \in L_j$ for which $\text{Adjacent}(l_{i,u}, l_{j,v}) = \text{true}$.

4) $d(r,s)$ is the feature distance between pixels r and s .

5) $f_{i,j} = |L_i^j|$, the approximated length of the common boundary between S_i and S_j

6) $\forall l_{i,u} \in L_i^j$, mean boundary distance between segments S_i and S_j , $CT_{i,j} = \frac{1}{3} \frac{1}{f_{i,j}} \sum d(l_{i,u}, l_{j,v})$, where $\exists u \in L_i, v \in L_j$ such that $\text{Adjacent}(l_{i,u}, l_{j,v}) = \text{true}$.

7) $d_{\text{weakness}} = \alpha \times d_{\text{max}}$, where $0 < \alpha < 1$ is the weakness control factor that decides what proportion of d_{max} is the allowable weakness between segments in order to allow a merge to take place.

In this step we check only neighbour $\{TL\}$ at every scan position, having observed faster processing times without significant degradation in quality. Since we proceed from the main segmentation step, no initialisation is required for the region reduction post-processing phase, which is given by the following:

- I. *Region reduction, $\forall t \in \text{Ind}$, 1-neighbour scan $\{TL\}$:*
 - (i) $\text{hasMerged} = \text{false}$.
 - (ii) $i = \text{Label}(t)$.
 - (iii) calculate $C_{u,v}$ as $CT_{i,j} | S_j \in B_i$.
 - (iv) if $C_{u,v} \leq d_{\text{weakness}}$, do $\text{Merge}(S_u, S_v)$ as follows:
 - a) $k = k + 1$ and $m = m + 1$.
 - b) $P^k = (P^{k-1} \cup \{S_m\}) \cap \{S_u, S_v\}$.
 - c) calculate D_m from \underline{D}_u and \underline{D}_v .
 - d) $B_m = (B_u \cup B_v) \cap \{S_u, S_v\}$.
 - e) $\forall S_j \in B_m, B_j = (B_j \cup \{S_m\}) \cap \{S_u, S_v\}$.
 - f) $\text{hasMerged} = \text{true}$.
 - (v) if $\text{hasMerged} = \text{true}$, repeat from step I.

Next we reduce the number of small segments by performing a single iteration of the main region merging algorithm via a full 8-neighbour scan (to avoid missing possible merge

pathways) by setting $d_{\text{curr}} = \infty$ and considering $i = \text{Label}(t)$ in a row-wise scan only for segments S_i such that the segment size $|S_i|$ satisfies the condition $|S_i| \leq \gamma \times \frac{|\text{Ind}|}{|P^{k-1}|}$, where the total image size is represented by the number of pixels contained $|\text{Ind}|$ and the control parameter γ decides the smallness to be merged in terms of some fraction of the ratio between image size and the number of segments obtained thus far.

Additionally, to deal with completely enclosed regions, we use a second weakness control factor β to reduce the cost of merging such regions. The main region merging process can be repeated for a single iteration again via a full 8-neighbour scan after setting $d_{\text{curr}} = \beta \times d_{\text{max}}$ and considering $i = \text{Label}(t)$ in a row-wise scan only for segments S_i in which $|L_i^j| = |L_i|$.

IV. RESULTS

Next we set values for the parameters $d_{\text{max}}, d_{\text{step}}, \alpha, \beta, \gamma$ used in our quantitative evaluation (see Section IV) experiments. Experimental observations over large sets of test data have suggested the following sets of parameter values to produce robust results across a wide variety of inputs:

$$d_{\text{max}} = 35, d_{\text{step}} = 3, \alpha = 0.55, \gamma = 2, \beta = 10.$$

Segmentation resolution is inversely proportional to d_{max} , α , β and γ . Segmentation speed is directly proportional to d_{step} , which also affects the evolution of the segmentation map by altering the affinity to merge regions.

While there have been many segmentation evaluation methods and metrics proposed over the years, only a few have been commonly used for the purposes of explicit comparison against other methods. Four of these metrics are the probabilistic Rand index (PRI) [29], variation of information (VOI) [22], global consistency error (GCE) [21] and boundary displacement error (BDE) [18], each having established figures for results using important segmentation methods. We use these four measures to quantitatively evaluate our segmentation results against the figures reported in [19], [25], [12], [11].

The comparisons are with respect to the publicly available Berkeley Segmentation Dataset (BSDS) [21]. The segmentation methods compared against are the following: *Average Tree Partitioning* (ATP) [35], *Contour Based Segmentation* (GPB-UCM) [5], *Simple Linear Iterative Clustering* (SLIC) [1], *Segmentation From a Soft Boundary Map* (SFSBM) [24], *K-Means clustering and Graph Cut* (KmsGC) [20], *Felzenszwalb & Huttenlocher Graph-based* (FH) [16], *Mean Shift* (MS) [13], *Normalised Cuts* (NC) [30], *Multiscale NCut* (MNC) [14], *Markov Chain Monte Carlo* (MCMC) [33], *Fusion of Clustering Results* (FCR) [23], *Compression-based Texture Merging* (CTM) [36], *Ultrametric Contour Maps* (UCM) [4], *Texture and Boundary Encoding-based Segmentation* (TBES) [25], *Hierarchical Markov Clustering* (HMC) [19], *Multiclass Spectral Clustering* (MCSpec) [37], *Normalised Tree Partitioning* (NormTree) [34], *Blobworld* (BW) [9], *Swendsen-Wang Cuts* (SWC) [32]. We abbreviate our semi-greedy adaptive-threshold method as SGAT. The quantitative comparisons are shown in Figure 3.

Additionally, since we are comparing segmentation methods based on values for four different metrics, we propose a way to obtain a single performance indicator from multiple

metrics. This performance indicator uses the results for human segmentation as a baseline. There are some properties we must be careful to preserve in the indicator. In some evaluation metrics lower values show better performance while for others higher values show better performance. We therefore use a scheme that adds a positive term to the performance indicator for metric results better than the baseline and penalises metric results worse than the baseline. We must also take into account the fact that different metrics produce typical values in varying numeric ranges, and thus differences in some metrics may have less significance than differences in others. To counteract this we normalise each metric result by the relative importance of that metric with respect to the complete set of metrics being considered.

Some desirable properties of the performance indicator are:

1) the indicator should have a zero value when the baseline is compared against itself.

2) the indicator should have a specific constant value when the same pair of human and algorithm results are being compared, independent of how many other algorithms are being compared against.

We call this new indicator the relative performance (RP), which is defined next. Let H_i and A_i be the baseline (human) and challenger (algorithm) results for metric i for n different metrics, also let $\lambda_i = 1$ when higher values are better and $\lambda_i = -1$ when lower values are better for metric i . Then the relative weight of each metric according to the baseline is given by:

$$W_i = \frac{H_i}{\sum_{j=0}^n H_j} \quad (1)$$

Then RP is defined as:

$$RP = \frac{1}{n} \sum_{i=0}^n \frac{\lambda_i \times (H_i - A_i)}{W_i} \quad (2)$$

As we can see from the RP column in Figure 3, this relative performance indicator allows us to compare overall performance for an algorithm across several metrics with respect to a set of baseline values for each metric. Higher values (closer to zero) are better. A positive value for RP would indicate that the algorithm has performed better than the baseline. We can see that, with respect to the available data, the proposed SGAT segmentation scheme produces the best overall performance across the metrics considered.

All experiments were run on a 2.26 Ghz dual core laptop computer. Average segmentation frame rates on the 481×321 BSDS images were 0.5 to 1 fps depending upon image complexity, for both region merging and region reduction phases.

We now move to a qualitative visual analysis of the SGAT segmentation results. We see from Figure 4 that important object boundaries and edges are found in each example, even in difficult cases such as the gorilla and snake images, even though the contour of the gorilla has been contorted by a small area of the background. We also observe that this segmentation

Algorithms	PRI \uparrow	VoI \downarrow	GCE \downarrow	BDE \downarrow	RP \uparrow
Human	0.8754	1.1040	0.0797	4.9940	0
SGAT	0.7946	3.5026	0.1396	5.0237	-5.33
CTM $\gamma=0.2$	0.7617	2.0236	0.1877	9.8962	-5.82
GPB-UCM	0.8183	1.547	0.1911	13.04	-6.13
FCR	0.7882	2.3035	0.2114	8.9951	-6.42
FH	0.7841	2.6647	0.1895	9.9497	-6.86
ATP	0.8039	2.021	0.2066	13.77	-7.52
NC	0.7229	2.9329	0.2182	9.6038	-7.92
MS	0.7550	2.4770	0.2598	9.7001	-8.08
MNC	0.7559	2.4701	0.1925	15.10	-8.49
KmsGC	0.7712	2.5616	0.2932	9.1225	-8.72
NormTree	0.7521	2.4954	0.2373	16.30	-9.95
MCSpec	0.7357	2.6336	0.2469	15.40	-10.10
HMC	0.7816	3.8700	0.3000	8.9300	-10.87
SLIC	0.7287	2.935	0.2738	18.93	-12.44
TBES	0.807	1.705	-	-	-
SFSBM	0.787	-	-	-	-
UCM	0.77	2.11	-	-	-
MCMC	0.768	2.261	-	-	-
BW	0.7138	2.6295	-	-	-
SWC	0.7644	3.0266	-	-	-

Fig. 3. Quantitative comparison of SGAT segmentation results with other methods. Average performance on the BSDS shown. Arrows next to metric name indicate direction of desired values. Figures not available are marked as '-'. The five best values for each measure are shown in bold.

method tends to retain dense groups of segments in highly textured areas indicating the complexity of the image over those areas. In the case of the bear image, the bear itself has been oversegmented and is not represented as a single object, which is due to textures and shades present on the fur of the bear. The overall visual quality of the results appear fairly consistent with human perceptual interpretation of these images. However the usefulness of segmentation results is best determined when applied to a larger cognitive task, different applications requiring varying levels of resolution for different purposes.

V. DISCUSSION

In this paper we have presented a novel segmentation method using a semi-greedy threshold-adaptive (SGAT) region merging scheme and region reduction mechanisms to refine the segmentation. The proposed method is fast and both the quantitative and qualitative results demonstrate its effectiveness. We have further proposed a new indicator of performance summary, the relative performance (RP), over several different commonly used evaluation metrics compared to a set of baseline results. This indicator allows us to easily compare computational and human segmentations.

REFERENCES

- [1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(11):2274–2282, 2012.
- [2] R. Adams and L. Bischof. Seeded region growing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16:641–647, June 1994.
- [3] M. Andreetto, L. Zelnik-Manor, and P. Perona. Non-parametric probabilistic image segmentation. In *IEEE Int'l Conf. Comp. Vis. (ICCV)*, pages 1–8, 2007.
- [4] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. In *IEEE Conf. Comp. Vis. Pattern Recog. (CVPR)*, pages 2294–2301, 2009.
- [5] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, 2011.

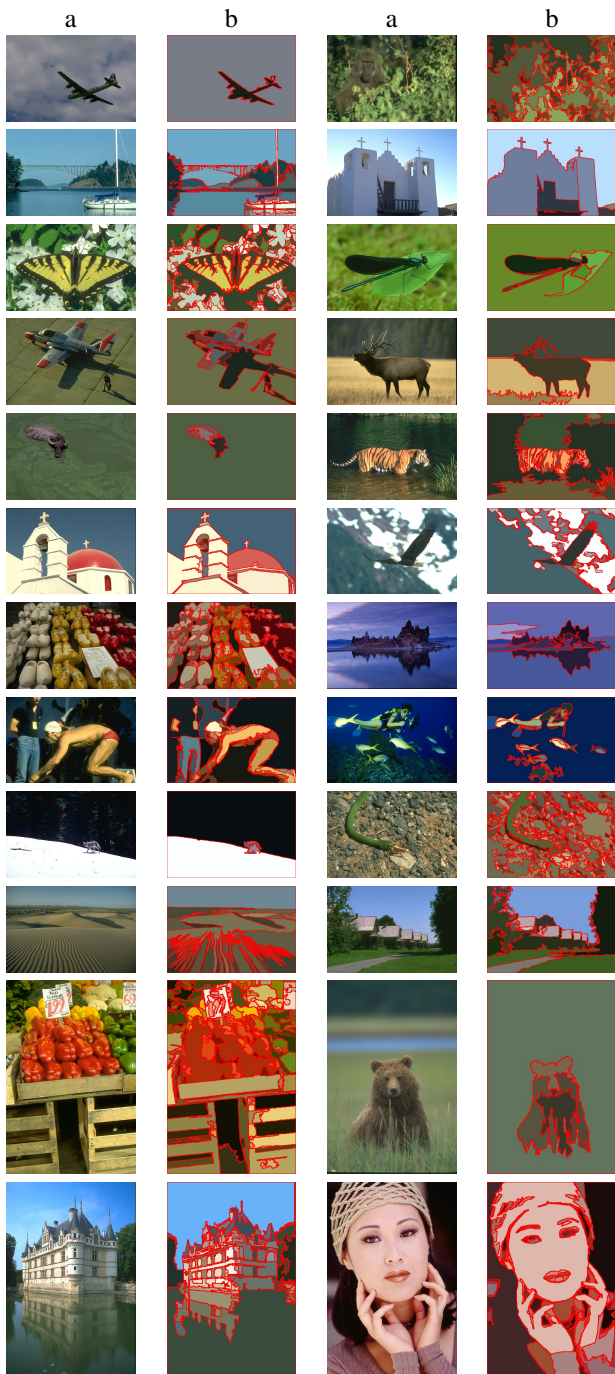


Fig. 4. Segmentation results on some BSDS images using SGAT. a) Original, b) Segmented

- [6] J.-M. Beaulieu and M. Goldberg. Hierarchy in picture segmentation: A stepwise optimization approach. *IEEE Trans. Patt. Anal. Mach. Intell.*, 11:150–163, February 1989.
- [7] C. R. Brice and C. L. Fennema. Scene analysis using regions. *Artif. Intell.*, 1(3):205–226, 1970.
- [8] F. Calderero and F. Marques. Region merging techniques using information theory statistical measures. *IEEE Trans. Img. Proc.*, 19:1567–1586, 2010.
- [9] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Trans. Patt. Anal. Mach. Intell.*, 24:1026–1038, 2002.
- [10] Y.-L. Chang and X. Li. Adaptive image region-growing. *IEEE Trans.*

- Img. Proc.*, 1994.
- [11] S. Chen, L. Cao, Y. Wang, J. Liu, and X. Tang. Image segmentation by MAP-ML estimations. *IEEE Trans. Img. Proc.*, 19:2254–2264, 2010.
- [12] M. Cho and K. M. Lee. Authority-shift clustering: Hierarchical clustering by authority seeking on graphs. In *IEEE Conf. Comp. Vis. Patt. Recog. (CVPR)*, pages 3193–3200, 2010.
- [13] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, 2002.
- [14] T. Cour, F. Benezit, and J. Shi. Spectral segmentation with multiscale graph decomposition. In *IEEE Conf. Comp. Vis. Patt. Recog. (CVPR)*, pages 1124–1131 vol.2, 2005.
- [15] D. S. Farin. *Automatic Video Segmentation Employing Object/Camera Modeling Techniques*. PhD thesis, Technische Universiteit Eindhoven, 2005.
- [16] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *Int'l J. Comp. Vis.*, 59:167–181, 2004.
- [17] C. Fredembach and G. Finlayson. Path based colour image segmentation. In *Eur. Conf. Col. in Graph., Imag. and Vis. (CGIV)*, pages 382–386, 2006.
- [18] J. Freixenet, X. Muñoz, D. Raba, J. Martí, and X. Cufí. Yet another survey on image segmentation: Region and boundary information integration. In *Eur. Conf. Comp. Vis. (ECCV)*, pages 408–422 Part III, 2002.
- [19] R. Hedjam and M. Mignotte. A hierarchical graph-based markovian clustering approach for the unsupervised segmentation of textured color images. In *IEEE Int'l Conf. Img. Proc. (ICIP)*, pages 1357–1360, 2009.
- [20] B. Liang and J. Zhang. KmsGC: An unsupervised color image segmentation algorithm based on K-means clustering and graph cut. *Hindawi Math. Problems Eng.*, page 13, 2014.
- [21] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Int'l Conf. Comp. Vis. (ICCV)*, pages 416–423 vol.2, 2001.
- [22] M. Meilă. Comparing clusterings: an axiomatic view. In *Int'l Conf. Mach. Learning (ICML)*, pages 577–584, 2005.
- [23] M. Mignotte. Segmentation by fusion of histogram-based K-means clusters in different color spaces. *IEEE Trans. Img. Proc.*, 17(5):780–787, 2008.
- [24] M. Mignotte. A non-stationary mrf model for image segmentation from a soft boundary map. *Patt. Anal. Applic.*, 17(1):129–139, 2014.
- [25] H. Mobahi, S. Rao, A. Y. Yang, S. S. Sastry, and Y. Ma. Segmentation of natural images by texture and boundary compression. *CoRR*, abs/1006.3679, 2010.
- [26] R. Nock and F. Nielsen. Statistical region merging. *IEEE Trans. Patt. Anal. Mach. Intell.*, 26:1452–1458, 2004.
- [27] S. Paris and F. Durand. A topological approach to hierarchical segmentation using mean shift. *IEEE Conf. Comp. Vis. Patt. Recog. (CVPR)*, 0:1–8, 2007.
- [28] A. K. Qin and D. A. Clausi. Multivariate image segmentation using semantic region growing with adaptive edge penalty. *IEEE Trans. Img. Proc.*, 19:2157–2170, 2010.
- [29] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.
- [30] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 22:888–905, 2000.
- [31] R. W. Taylor, M. Savini, and A. P. Reeves. Fast segmentation of range imagery into planar regions. *Comp. Vis. Graph. Img. Proc.*, 45:42–60, 1989.
- [32] Z. Tu. An integrated framework for image segmentation and perceptual grouping. *IEEE Int'l Conf. Comp. Vis. (ICCV)*, 1:670–677, 2005.
- [33] Z. Tu, S.-C. Zhu, and H.-Y. Shum. Image segmentation by data driven Markov chain Monte Carlo. In *IEEE Int'l Conf. Comp. Vis. (ICCV)*, pages 131–138 vol.2, 2001.
- [34] J. Wang, Y. Jia, X.-S. Hua, C. Zhang, and L. Quan. Normalized tree partitioning for image segmentation. In *IEEE Conf. Comp. Vis. Patt. Recog. (CVPR)*, pages 1–8, 2008.
- [35] J. Wang, H. Jiang, Y. Jia, X.-S. Hua, C. Zhang, and L. Quan. Regularized tree partitioning and its application to unsupervised image segmentation. *IEEE Trans. Img. Proc.*, 23(4):1909–1922, 2014.
- [36] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry. Unsupervised segmentation of natural images via lossy data compression. *Comput. Vis. Img. Underst.*, 110:212–225, 2008.
- [37] S. X. Yu and J. Shi. Multiclass spectral clustering. In *IEEE Int'l Conf. Comp. Vis. (ICCV)*, pages 313–319 vol.1, 2003.